Structure-Aware Deep Learning for Product Image Classification

ZHINENG CHEN, Institute of Automation, Chinese Academy of Sciences, China
SHANSHAN AI and CAIYAN JIA, Beijing Jiaotong University, China

Automatic product image classification is a task of crucial importance with respect to the management of online retailers. Motivated by recent advancements of deep Convolutional Neural Networks (CNN) on image classification, in this work we revisit the problem in the context of product images with the existence of a predefined categorical hierarchy and attributes, aiming to leverage the hierarchy and attributes to improve classification accuracy. With these structure-aware clues, we argue that more advanced deep models could be developed beyond the flat one-versus-all classification performed by conventional CNNs. To this end, novel efforts of this work include a salient-sensitive CNN that gazes into the product foreground by inserting a dedicated spatial attention module; a multiclass regression-based refinement that is expected to predict more accurately by merging prediction scores from multiple preceding CNNs, each corresponding to a distinct classifier in the hierarchy; and a multitask deep learning architecture that effectively explores correlations among categories and attributes for categorical label prediction. Experimental results on nearly 1 million real-world product images basically validate the effectiveness of the proposed efforts individually and jointly, from which performance gains are observed.

CCS Concepts: • **Computing methodologies** → *Computer vision tasks*;

Additional Key Words and Phrases: Image classification, category hierarchy, convolutional neural network, multi-class regression, multi-task learning

ACM Reference format:
Zhineng Chen, Shanshan Ai, and Caiyan Jia. 2019. Structure-Aware Deep Learning for Product Image Classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 1s, Article 4 (January 2019), 20 pages.
https://doi.org/10.1145/3231742

4

1 INTRODUCTION

Just as Yahoo! established the order of the Internet in its infancy, using categories like News, Sports, and Music to manage the rapidly increasing number of online documents and resulting in a much better experience for browsing and searching, categorization also plays an indispensable role in organizing online products. An immense amount of products covering almost all aspects of our lives are made available through online retailers such as *Taobao*, *Amazon*, and others due to the

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61772526 and 61473030, and the National Key R&D Plan of China under Grant No. 2017YFB1002804.
Authors' addresses: Z. Chen, Institute of Automation, Chinese Academy of Sciences, No. 95, Zhongguancun East Road, Haidian District, Beijing, China, 100190; email: zhineng.chen@ia.ac.cn; S. Ai, Beijing Jiaotong University, No. 3, Shangyuancun Road, Haidian District, Beijing, China, 100044; email: 15120384@bjtu.edu.cn; C. Jia (Corresponding author), Beijing Jiaotong University, No. 3, Shangyuancun Road, Haidian District, Beijing, China, 100044; email: cyjia@bjtu.edu.cn.

1551-6857/2019/01-ART4 $15.00
https://doi.org/10.1145/3231742

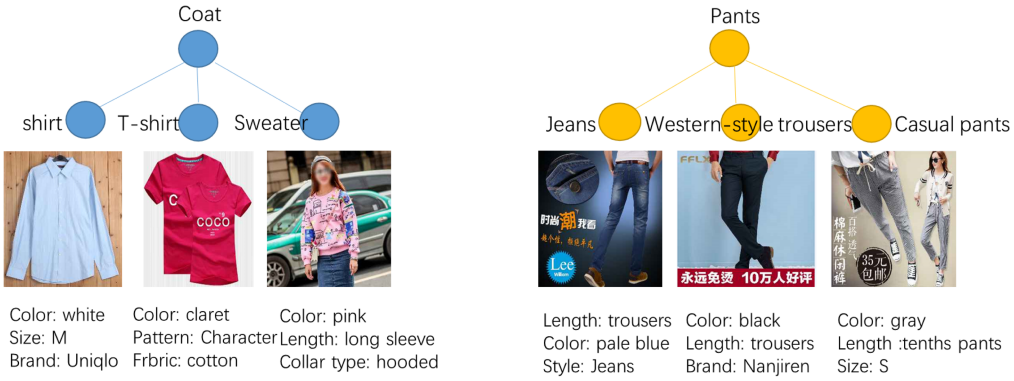

Fig. 1. An illustrative example of the two-layer categorical hierarchy of *Taobao*, where each product is tagged with two categorical labels and/or a few descriptive attributes.

maturity of the Internet and the prevalence of e-shopping. For example according to statistics from *360pi*[1], Amazon sold more than 30 million products online as of May 2016. Product categorization provides a practical way to organize these huge numbers of products by partitioning them into many meaningful subsets. Good categorization benefist both the customer and the retailer. On one hand, customers can conveniently find products they interested in, thus bringing a better shopping experience. On the other hand, business statistics like sales data can accumulate at a finer grained granularity (i.e., category by category), which is crucial to support better business decisions. Moreover, categorization is also a building block for a number of customer-oriented services such as product display, recommendation, search, and the like.

To organize these huge numbers of products efficiently, almost all online retailers use a hierarchy with predefined categories rather than a flat categorization. For example, *Taobao* employs a two-layer hierarchy to manage its products. Each product is characterized by a root category (e.g., *shoes*) and a fine-grained leaf category (e.g., *running shoes*) belonging to this root category. The categories correspond to disjoint partitions of products at different granularities. In addition, the product is also characterized by some descriptive attributes (e.g., *color*, *brand*, *fabric*, etc.). Figure 1 provides a vivid illustration. Unlike a category that corresponds to a distinct data partition, different products can be described by the same attribute with various states, and moreover, the attribute pool varies by categories (e.g., *fabric* is available for clothes but absent for electronic devices). The categories and attributes are two kinds of structural labels that describe a product from different angles. It is worth mentioning that retailers other than *Taobao* may have different categorizations. For example, *JingDong*, one of the most popular online retailers in China, uses a deeper hierarchy in which some attributes (e.g., *brand*) are also nodes in the hierarchy. However, these nodes can be eventually classified as two kinds of labels by determining whether they correspond to a distinct data partition with respect to the product repository as the metric.

Given categorization, it becomes an essential demand to automatically predict the categorical and attributive labels of online products, especially for categorical labels. Because people are continuously uploading products to online retailers, labels are an essential but laborious task when done manually. Alternatively, visual-based automatic label inference can implement this task efficiently and without bias and is thus highly desired. Similar topics have been intensively studied for years in the form of image classification, annotation, etc., where considerable progress has been reported, especially with recent deep learning-based solutions [5–7, 17, 21, 22, 27, 40, 41]. An

[1]https://markettrack.com/360pi-and-market-track.

example is the 1,000-category ImageNet classification competition. It reports a top-5 error rate of 25.8% in 2011 using methods based on handcrafted shallow features, while the error rate dropped considerably to 15.3% in 2012 using AlexNet, an 8-layer deep CNN that operates on image pixels directly [19]. Moreover, it rapidly declined to an incredible 3.5% in 2017 using a much deeper DenseNet [13]. Meanwhile, fine-grained image classification also gained substantial performance improvements [9, 32]. Compared with shallow-based methods, a CNN generally generates better discriminative region localization and feature learning, both of which are crucial in identifying objects with small visual variances. In these studies, the categories are treated equally and independently following a flat categorization scheme.

In this article, we are mainly interested in predicting categorical labels given a product image associated with a hierarchical category representation and/or a few attributes. We argue that this scenario is quite common for online retailers. Because of the diversity of products and their images, existing work on this field mostly focuses on classifying a subset of products, such as clothes [25, 39], shoes [26], and groceries [10]. Classifying products covering a wide range of categories remains a challenge yet to be fully explored. In addition, this problem also represents a distinctive image classification scenario compared with the aforementioned studies: Both the hierarchy and attributes represent some kinds of information which can be utilized to improve the classification performance intuitively. However, compared with flag categorization, which has been intensively studied, how to explore these structure-aware clues to maximize classification accuracy, especially for large-scale product classification, is still a field receiving little attention.

Motivated by this, in this work, we investigate the problem of large-scale product image classification with the existence of a predefined category hierarchy and attributes. Different from conventional CNNs that merely treat classification as a flat one-versus-all problem, we aim to leverage the hierarchy and attributes to further improve classification accuracy. We argue that more advanced CNNs could be developed based on these structure-aware clues. To this end, we worked on the following three directions.

First, since the product foreground often occupies a small portion of an image and the background usually provides less information for classification, we develop a salient-sensitive CNN by inserting a spatial attention module into existing CNN architectures. The module is trained along with the whole network. It implements a salient-sensitive spatial selection by providing a dynamic weight matrix which indicates importance of the corresponding spatial grids of a given image. By assigning large weights, the product foreground can play a more important role in category prediction.

Second, we propose a Multiclass Regression (MCR)-based refinement to derive more accurate prediction scores. Our idea is that multiple CNNs can be trained straightforwardly by leveraging the hierarchy (i.e., train each CNN by merely considering the categorical labels of a certain layer in the hierarchy; e.g., all leaf categories). Prediction scores from these CNNs are correlated to each other through the inclusion relationship across these categories. MCR-based refinement can be viewed as a data-driven method that encodes correlations among classification scores from different CNNs, from which more accurate predictions are expected.

Third, we devise a multitask deep learning architecture to implement the classification. The architecture takes the prediction of categories and attributes as different tasks in a single network. The network architecture and its loss function are properly designed to optimize the tasks simultaneously. Unlike MCR-based refinement that implements classification via two disjoint steps (i.e., CNN classification and MCR-based refinement), multitask learning takes these mutually correlated tasks as a whole and uses backpropagation to reinforce them end-to-end.

In experiments, we carry out extensive evaluations to quantitatively analyze the three efforts on nearly 1 million real-world product images. Experimental results validate the effectiveness of

the efforts individually and jointly, from which consistent performance gains are observed. Our studies give valuable insights into applying structure-aware knowledge to the task of large-scale image classification, especially in the context of product image classification.

This article extends significantly our preliminary work [1], which implements large-scale product classification via spatial attention-based CNN learning and MCR-based refinement. Compared with that earlier work [1], our novel efforts here include the utilization of attributes, the end-to-end trainable multitask CNN architecture, and more comprehensive experiments and analyses. The main contributions of this article come from three aspects.

- We give an in-depth investigation on the task of large-scale product image classification. Differing from existing studies, we emphasize leveraging structure-aware knowledge in the form of a predefined category hierarchy and attributes to improve classification performance of popular CNN models.
- We propose three novel efforts to implement the structure-aware product classification. They are salient-sensitive CNN, MCR-based refinement, and multitask deep learning, which discover correlations among the categories and/or attributes from different angles. Our studies provide insights into how to utilize these structural clues.
- We conduct experiments on nearly 1 million real-world product images covering more than 100 categories. The proposed efforts are evaluated individually and jointly, from which performance gains of the efforts are quantitated.

The remainder of the article is organized as follows. In Section 2, we describe existing work related to our study. Section 3 describes in detail our techniques to model the hierarchical categories and attributes to assist the classification. The experimental results and analyses are shown in Section 4. Finally, we conclude this article and point out some future work in Section 5.

2 RELATED WORK

Image classification is a long-standing research topic in computer vision and multimedia. Despite facing great challenges, many research efforts have been made in the past decades especially in the era of deep learning. Based on the coverage of the image to be classified, we can broadly categorize existing studies into large-scale generic classification and fine-grained classification.

2.1 Large-Scale Generic Classification

Large-scale generic image classification refers to identifying images covering a wide range of objects or scenes in natural life. The task is challenging as it always deals with a large number of categories with varying visual appearances. To promote studies on this direction, Li et al. launched the ImageNet classification competition whose objective was to classify a given image to one of 1,000 predefined categories [8]. In the initial years of the competition, it was common to use sophisticated methods built on top of high-dimensional handcrafted shallow features. However, the best performing method still returned a top-5 error rate of more than 25% [33]. In 2012, the invention of AlexNet brought the study to the era of CNNs, where a substantially lower error rate of 15.3% was reported using this 8-layer CNN, showing that deep models built on top of big data could produce powerful nonlinear feature representations and modeling abilities. From then on, several more efficient CNNs were proposed in a few years. For example, in 2014, Simonyan et al. and Szegedy et al. devised VGG [35] and GoogleNet [36], respectively. The VGG-16 reported a phenomenal top-5 accuracy of 92.7%. On the other hand, GoogleNet invented a module called *inception* that approximated a sparse CNN with a conventional dense construction. It achieved a 6.7% top-5 error rate on ImageNet using a 22-layer network. With the network going deeper, He et al. proposed ResNet [12]. The 152-layer ResNet reported a top-5 accuracy of 95.5%, and its performance

was beyond human vision in this particular task. Recently, an almost inconceivable top-5 error rate of 3.5% was obtained using DenseNet [13]. It used an important variant on top of ResNet in that each layer took all preceding feature maps as input, generating denser feature representation. These studies emphasized devising powerful CNN architectures to improve the performance of large-scale classification. However, these networks were designed as the flat one-versus-all structure where all categories were treated equally and independently. Correlations between them thus could not been well utilized, although it was shown to be useful [16, 44]. For example, the HD-CNN [44] employed two modules to generate prediction scores for easy to separate and difficult to distinguish categories, respectively, where correlations among categories were explored to facilitate the classification of difficult-to-distinguish ones.

Other studies employed hierarchical concepts to alleviate the difficulties caused by the flat one-versus-all classification. Xie et al. [42] proposed probabilistic visual concept trees and applied this in visual concept detection. By organizing concept classifiers into a hierarchy, only a small set of relevant classifiers was evaluated for prediction. It showed, for the first time, that large-scale image classification could benefit from hierarchically structured categories. To reduce the risk of propagating classification errors from higher layers of the hierarchy to lower layers, Zhu et al. [49] utilized sibling categories and their children to collaboratively rectify unreliable classifications. In Lei et al. [20], the HLMMs method was developed to implement large-scale image classification effectively through the hierarchical learning of large-margin metrics. Mai et al. also investigated techniques to make the categorical tree more balanced [28]. Recently, Zhang et al. proposed a novel image representation that clustered images hierarchically to explore their correlations; image classification experiments on several benchmarks demonstrate its effectiveness [47]. Hierarchical structures were shown to be useful in mining correlations among categories. However, how to leverage the structure using deep learning is less studied, especially in the context of product image classification.

2.2 Fine-Grained Classification

Unlike generic classification, fine-grained image classification focuses on classifying among categories that are both visually and semantically similar (e.g., car models, bird species, etc.). The task is challenging because the different categories only exhibit marginal visual differences. Accordingly, existing studies mostly focus on two subtasks that can amplify these differences: discriminative part localization and feature learning.

In *discriminative part localization*, early works mainly focused on leveraging extra object bounding boxes and part annotations to localize discriminative regions [14, 23]. They required heavy involvement of manual labeling efforts and thus were difficult to scale up to large-scale scenarios. An emerging trend is to use unsupervised approaches to locate the regions. For example, by getting filter responses from CNNs that respond to specific patterns, deep filter responses [48] and multigrained descriptors [37] were picked to learn a set of part detectors. Jaderberg et al. [15] proposed a spatial transformer network that could dynamically and spatially transform an image for more accurate classification. Fu et al. [9], the authors took one step further and developed a novel recurrent attention CNN to recursively learn discriminative region localization and region-based feature representation at multiple scales in a mutually reinforced way.

On the other hand, learning discriminative features for located regions is also crucial for recognizing fine-grained classes. Since features extracted from a CNN perform significantly better than handcrafted features, it is prevalent now to use powerful CNN architectures as feature extractors. Based on this, a bilinear structure was devised [24] that used two independent CNNs to compute the pairwise feature interactions and capture image local differences. Experiments on bird classification demonstrate its effectiveness. Moreover, Zhang et al. [48] proposed to unify a CNN with

spatially weighted representations using a Fisher Vector [30]. The method also showed superior results on both bird [38] and dog [18] datasets.

These studies mainly experimented on specific objects like birds and dog. There also were some studies focusing on search and classification of products. For example, Huang et al. [26] proposed the Circle and Search system. Given a shoe image, it first developed an attribute-aware shoe detection model and then combined the model with retrieval techniques to get the search result. George et al. proposed a pre-exemplar multilabel product classification method for simultaneous product recognition and localization. Its effectiveness was validated on a grocery dataset [10]. Liu et al. devised a FashionNet to learn clothing features by jointly predicting clothing attributes and landmarks. Based on it, they implemented effective clothes recognition and retrieval [25]. In Chai et al. [3], the authors used the Campana-Keogh descriptor to predict the labels of top-ranked product images and re-rank them. The method was validated on a product dataset containing 100,000 images covering 25 categories. In these studies, the kinds of product employed is often too narrow in coverage. Moreover, some of them heavily depend on a product's domain knowledge (e.g., shoe shape and attributes [26]), which are difficult to extend to other kinds of product.

Compared with the aforementioned studies, in this work we focus on classifying all products that could appear in online retailers. Typically, this case covers a wide range of products simultaneously associated with a number of categories that are hierarchically structured and some descriptive attributes. The two kinds of labels are both predefined. There are rich correlations among them intuitively. Unfortunately, it is unclear how to leverage this structural knowledge to enhance classification accuracy. We believe that this particular scenario presents a meaningful research topic that has been little explored.

3 THE PROPOSED METHOD

3.1 Problem Formulation

We aim to study methods for product image classification with the presence of a predefined category hierarchy and a set of descriptive attributes. Let the hierarchy be $\Omega = \{C^1, C^2, \ldots, C^h\}$, where h is depth of the hierarchy, C^i denotes the set of categories at the ith layer in the hierarchy, and c^i_j is the jth category in it. Further denote the set of attributes as $D = \{d^1, d^2, \ldots, d^k\}$. Given a product image I, the objective of classification is to find a sequence of labels $c_p = \{c^1_p, c^2_p, \ldots, c^h_p\}$, which satisfies

$$c^i_p = \underset{c \in C^i}{argmax} f_c(I), \tag{1}$$

where $c \in C^i$ is a category at the ith layer, $f_c(I)$ represents the prediction score that image I being classified to category c, and c^i_p is the predicted category at the ith layer. We further denote $c_g = \{c^1_g, c^2_g, \ldots, c^h_g\}$ as the ground truth categorical label. If $c^i_p = c^i_g$ always satisfies for $\forall i \in [1, h]$, it is said to be a correct classification; otherwise, it is an incorrect classification. In case that the categories are hierarchically partitioned and nonoverlapping at each layer (i.e., every C^i corresponds to a distinct segmentation of products at a certain granularity), the classification task can be simplified as requiring merely $c^h_p = c^h_g$, as it also gives predictions at all layers in the hierarchy.

Since the CNN is widely recognized as one of the most powerful models in large-scale image classification, a straightforward implementation of classification treats it as a flat one-versus-all problem by using a typical CNN as the classifier. For example, h CNNs, each denotes the classifier at a certain layer, is trained separately, and then predictions at different granularities are obtained accordingly. Although a decent performance is expected, we argue that this implementation not only ignores the characteristics of product images, but also does not fully make use of the

structure-aware knowledge conveyed (i.e., the category hierarchy and attributes). More specifically, not all spatial regions of a product image are equally important with respect to the category identification. For example, the foreground may occupy a small portion of an image, or there may be category-insensitive text and/or logos in an image. Properly taking into account these factors is likely to generate a more accurate prediction. On the other hand, categories at different layers annotate a product at different granularities. It is obvious that they are correlated to each other; for example, the prediction on one layer (e.g., *shoes*) can affect and is also affected by the prediction on another layer (e.g., *running shoes*). Moreover, attributes are also crucial descriptions toward product category recognition. However, these structure-aware clues are ignored entirely by the flat one-versus-all classification. How to utilize them to further improve classification accuracy still needs exploration. In the rest of this section, we will introduce our efforts in dealing with these issues in detail, which consists of three major aspects.

- A salient-sensitive CNN that equips the CNN with a spatial attention module to instruct the model to pay more attention to the product foreground in an image.
- An MCR-based refinement which uses multiclass regression to fuse the prediction scores from different CNNs. Each CNN is a distinct classifier in the hierarchy.
- A multitask CNN architecture for category prediction, which jointly utilizes the categories and attributes of the product to supervise the model learning, performing structure-aware deep learning in a smooth and unified way.

3.2 CNN with a Spatial Attention Module

In product images, the foreground is often somewhat prominent in order to quickly attract users' interest. It plays a dominate role in determining a product's category. However, it is common that the foreground merely occupies a small portion of an image along with the category-indistinct background. If the classification model could gaze into the product foreground rather than at the whole image, a higher classification performance is expected.

With this idea in mind, we investigate how to incorporate visual attention analysis to assist in CNN-based classification. There are two types of methods to implement it. The first one is sequentially salient object detection and CNN prediction. It first automatically locates the product foreground in an image, and then feeds the cropped foreground image to the CNN to get the classification. However, this method breaks the classification into two separate steps. It thus heavily depends on the effectiveness of salient object detection and has the risk that incorrectly cropped images may lead to even worse classification results. On the other hand, some recent papers encode attention analysis into the CNN (e.g., adding a dedicated attention module). For example, in Xu et al. [43], an attention-based model was introduced to automatically learn the content description of images, whereas in Yang et al. [45], the authors devised a spatial attention module to better grasp the prominent image region. Compared with the sequential scheme, encoding an attention module into the CNN implements attention learning in a smoother and more unified way. It can be viewed as a kind of soft voting that is less affected by incorrect foreground localization. Therefore in this work, we consider the latter scheme.

Concretely, we would like to insert a dedicated spatial attention module into existing CNNs. The module can learn the weight parameters of different image regions in a dynamic way. Taking VGG-16 as the example, similar to Yang et al. [45], we add a spatial attention module behind *pool5* (i.e., the fifth pooling layer which outputs 512 feature maps of size 13*13). To ensure a seamless insertion (i.e., one that does not affect structure of the rest of the network), the module consists of the following building blocks. First, it contains a convolutional layer with kernel size of 1*1 whose parameters are initialized with the standard Gaussian distribution. The layer is followed by

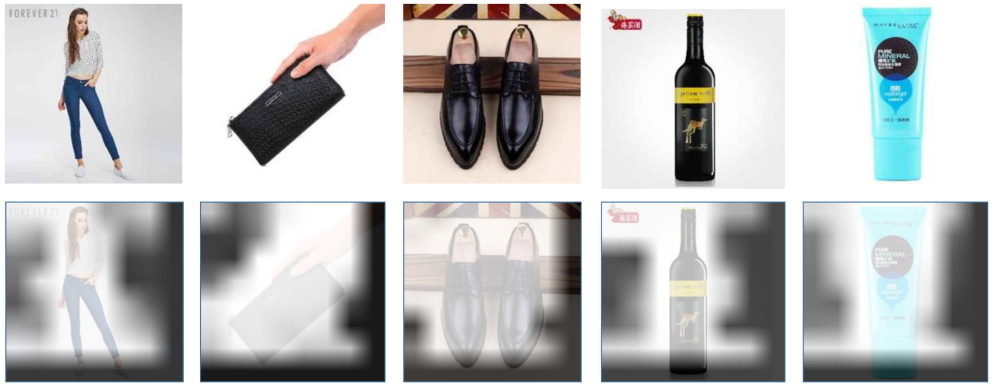

Fig. 2. Typical product images and their feature maps after salient-sensitive learning. Brightness of the feature map indicates the importance of the corresponding spatial grids in the image.

a sigmoid activation to rectify the values to $(0, 1)$. An additional spatial weight template of size 13*13 is learned for all the feature maps during the training process. Then, the attention learning is formulated as performing element-wise multiplication between the *pool5* feature maps and the weight template, which also outputs 512 feature maps of size 13*13.

Once the network is learned, the template can be viewed as a dynamic weight matrix that indicates the importance of the spatial grids of a given image. By assigning prominent and indistinctive grids with large and small weights, respectively, a salient-sensitive classification is performed accordingly. In Figure 2, we depict some product images and their feature maps after spatial attention learning. Prominent product regions are always highlighted. This method indeed instructs the network to focus more on the product foreground.

Note that other insertion positions are also possible for the spatial attention module, such as behind *pool4* or behind *conv5* (i.e., the fifth convolutional layer), etc. Our empirical studies show that slightly better results are obtained by inserting it behind more abstract pooling layers. Moreover, adaptions are essential when inserting the module behind a CNN layer with different sizes. For example, for a CNN layer with m feature maps of size $n*n$, adding a spatial attention module behind it can be simply done by setting the weight template as $n*n$ and then performing m times element-wise multiplication with the feature maps.

3.3 Classification Based on Multiclass Regression

With the classification model such as the salient-sensitive CNN developed in Section 3.2, we can predict categories at different granularities by constructing h CNNs, with each aiming to classify a different layer in the hierarchy. However, it is not optimal if we treat the h CNNs independently. There are correlations among categories at different layers that are not being explored. For example, if categories *shoes* and *running shoes* are returned with high probabilities by two different CNNs, the predictions are consistent and thus can reinforce each other (e.g., increasing the probabilities for both categories). On the contrary, if categories *coat* and *running shoes* are predicted with high probabilities, the predictions are conflicting such that their confidences should be challenged. The hierarchy inherently carries mutually correlated semantics that are valuable for better classification. However, they are not explored by separately modeled CNNs.

Motivated by this observation, we propose an MCR-based refinement to merge the classification scores obtained from different CNNs, aiming at leveraging the hierarchical relationships to reinforce their prediction scores. An illustrative flowchart of the proposed refinement is depicted in

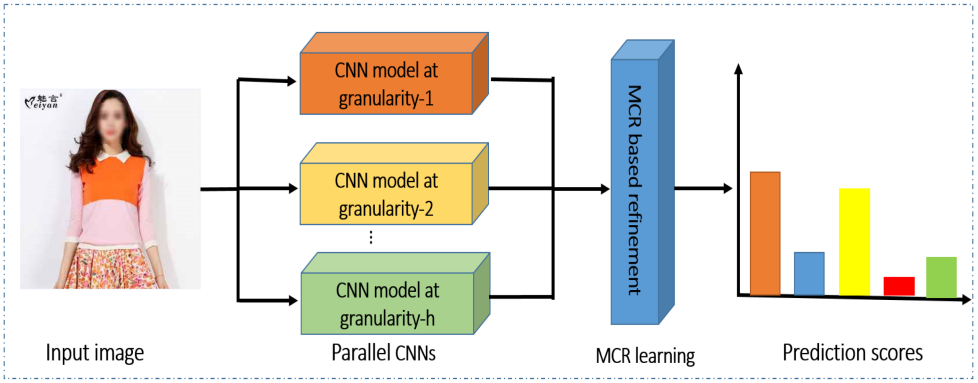

Fig. 3. An illustrative flowchart of Multiclass Regression (MCR)-based refinement. An MCR module is equipped behind h parallelized CNNs that perform the category classification at different granularities (i.e., granularity-1 to granularity-h.

Figure 3. Given a product image, it is independently fed into the h CNNs first for category prediction at different granularities. Then, the prediction scores are refined by MCR, where coherent and conflicting predictions are leveraged to adjust the scores, resulting in more accurate classification.

To formulate the MCR-based refinement, denote $\widehat{f_c}(I)$ as the prediction score that image I is being classified as category c before the refinement. Similar to Sánchez and Perronnin [34], we use the unnormalized prediction score rather than class posteriors of the softmax layer as $\widehat{f_c}(I)$, which approximates the differences among categories more realistically. Since $f_c(I)$, the refined score with respect to $\widehat{f_c}(I)$ is affected by some categories in both the same and different layers; further, let $R_c \subseteq \Omega$ be a set of q categories in the hierarchy that are semantically related to category c (e.g., categories describing visually and semantically similar products), and let $\hat{\mathbf{f}}_c(I)$ be the score vector corresponding to R_c from the h CNNs. We can formulate the refinement as

$$f_c(I) = P(c|\hat{\mathbf{f}}_c(I), \mathbf{W}) = \frac{exp(\mathbf{w}_c^T \hat{\mathbf{f}}_c(I))}{\sum_{t \in R_c} exp(\mathbf{w}_t^T \hat{\mathbf{f}}_c(I))}, \tag{2}$$

where $\mathbf{w}_c$ is a q dimensional weight vector that represents the degree of intimacy with other categories in R_c with respect to c. $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]$ is the complete weight matrix, and $n = |\Omega|$ is the number of distinct categories in the hierarchy. As explained in Equation (2), $f_c(I)$ is calculated based on the scores of preceeding CNNs on relevant categories and the learned weights.

Given a set of samples $X = \{x_1, x_2, \ldots\}$ with category labels $L = \{L_1, L_2, \ldots\}$, where $L_i = \{l_i^1, l_i^2, \ldots l_i^h\}$, the optimal weight matrix of $\mathbf{W}$, denoted as $\mathbf{W}^*$, can be obtained by

$$\mathbf{W}^* = \underset{\mathbf{W}}{argmin} - \sum_{x_i \in X} \sum_{j=1}^{h} logP\left(l_i^j | \mathbf{W}, \hat{\mathbf{f}}_c(x_i)\right) + \lambda \|\mathbf{W}\|^2, \tag{3}$$

where λ is a regularized parameter. Equation (3) is referred to as an L2-regularization MCR, which can be efficiently solved by the Quasi-Newton method. With the solved $\mathbf{W}^*$, $f_c(I)$ can be obtained accordingly by Equation (2).

A crucial issue in MCR-based refinement is how to determine R_c. Intuitively, setting $R_c = \Omega$ is not a wise choice as some categories are loosely correlated to c, which is likely to introduce noise if all of them are considered. On the other hand, setting $R_c = c$ is also meaningless as it totally ignores correlations among the categories. To trade off, we propose a heuristic to determine appropriate

R_c. Specifically, for a category c_i^j, we set

$$R_{c_i^j} = \left\{C^1, \ldots, C^{j-1}, T_i^j\right\}, \tag{4}$$

where T_i^j is composed of K categories in the jth layer (and its children categories in case the hierarchy is nonoverlapped partitioned) that mostly correlated to c_i^j. The K categories are determined by inspecting the learned weights (in absolute value) $\mathbf{W}$ from large to small. The reason for selecting these categories is twofold. First, categories in shallow layers are easier to distinguish than in deep layers in the hierarchy. Their classification scores are more reliable in general, so we take into account all of them. Second, larger weight can be roughly explained as more correlated. These categories are also informative in terms of classifying an image to that category. By considering semantically related categories, irrelevant noise is suppressed to a large extent. We empirically set K to 3 in this article. Preliminary results show that the proposed heuristic can lead to performance gains when compared with the strategy of setting $R_c = \Omega$.

Selecting a subset of categories as R_c also triggers an engineering problem when solving Equation (3) because the dimensions of weight matrix $\mathbf{W}$ vary by categories. Therefore, for uniformity and clarity, we fix the length of $\hat{\mathbf{f}}_c(I)$ to be $|\Omega|$ (i.e., all categories in the hierarchy). Accordingly, elements corresponding to $c \notin R_c$ are set to value 0 when optimizing Equation (3).

3.4 Classification Based on Multitask Deep Learning

In the MCR-based refinement, the CNNs and weight matrix (i.e., $\mathbf{W}$ in Equation (2)) are trained separately. There is no interaction between the two steps. However, the product image is annotated by multiple labels including several categories and/or some descriptive attributes in general. The classification can be regarded as a multilabel classification problem. Therefore, motivated by other works [2, 4, 11, 29, 31, 46], we propose to formulate the category classification as a multitask deep learning problem. Different from traditional CNN models that are trained with single label, multitask deep learning is composed of multiple tasks related to each other, and the objective is to classify all labels as correctly as possible by using an end-to-end trainable network. For example, in our case, the tasks include predicting the categories at different granularities and the attribute labels.

To achieve this goal, we modify the architecture of the CNN as follows. First, we pick out VGG-16 as the backbone. Since there are multiple classification tasks, we segment VGG-16 layers to public and private modules, respectively. All convolutional layers and the first fully connected layer constitute the public module shared by all the tasks. The module not only efficiently captures the low-level feature representations of product images, but it also mines the correlations between different tasks. On the other hand, the remaining deep layers (e.g., the second and the last fully connected layers) constitute the task-specific private module. It aims to learn the characteristics of different tasks individually. Different from the VGG-16, that has only one private module, the modified architecture has multiple private modules whose number is equal to the task branches, and each private module is merely related to one of the branches. Therefore, the number of outputted nodes of a private module is either the number of categories of a certain layer in the hierarchy or the number of distinct attributes with their possible states. Figure 4 illustrates a general structure of the proposed architecture, which is termed a multitask CNN in this work. In that network, we can choose all or part of the categorical layers and/or attributes to formulate the tasks.

In addition to the architecture, the loss function is also crucial for attaining good performance. Note that both category and attribute predictions can be regarded as classifications. We adopt the multinomial logistic loss as the loss function for all the modules. The overall loss function L is

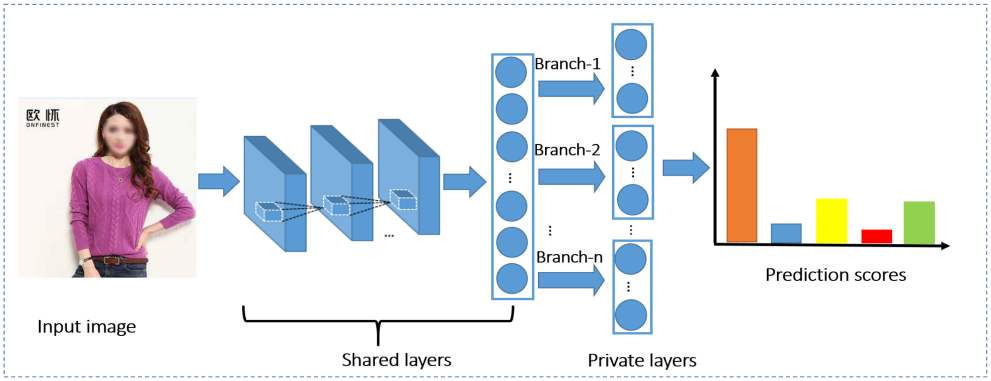

Fig. 4. Architecture of the proposed multitask CNN.

defined as

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{T}\lambda_i L_{i,n}, \tag{5}$$

where N is the total number of training images, T is the number of different tasks considered, and $\lambda_i \in [0, 1]$ is the loss weight of the ith task. There are two ways to adjust the network parameters in the training stage. For the public module, the losses from different tasks are linearly combined at first, and, based on this, parameters of public module (i.e., the first 14 layers for VGG-16) are updated accordingly. For the private module, the loss of each task is used to update the parameters of its private module individually. Let $\hat{P}^i_{n,y}$ be the prediction score of an image I_n being classified to its ground truth label y at the ith task, where the score is obtained from the softmax activation. For this case, the ith task is category classification, and the $L_{i,n}$, is given by

$$L_{i,n} = \log\left(\hat{P}^i_{n,y}\right). \tag{6}$$

In the case that the task is attribute prediction, the loss is defined differently as an attribute may have several to dozens of different states. Therefore attributes could not be learned by using the same strategy as categories. To learn the attributes, suppose that there are k attributes in total and the ith attribute is labeled by a state value $y^i \in \{0, 1, \ldots, m_i\}$, where value 0 denotes the attribute is absent (i.e., not labeled), m_i is the number of possible states of this attribute, and $y^i = l$ means the attribute is labeled as the lth state. We formulate attribute learning as an additional multi-task structure as follows. First, a fully connected layer with $\sum_{i=1}^{k} m_i$ nodes is developed. Then k sub-task layers, each with m_i nodes, are plugged behind this layer. The two-layer structure constructs a dedicated task branch for attribute prediction. Given an image I_n with attribute labels $\{y^1_n, y^2_n, \ldots, y^k_n\}$ where y^j_n is a distinct state of the jth attribute, let γ_j denote the weight parameter of the jth attribute and $L_{i,j,n}$ represent the loss for the jth attribute of I_n calculated by Equation (6). The $L_{i,n}$ is defined as follows:

$$L_{i,n} = \sum_{j=1}^{k} S\left(y^j_n \neq 0\right)\gamma_j L_{i,j,n}, \tag{7}$$

where $S(\cdot)$ is an indicator function that equals 1 if the contained condition is satisfied and equals 0 if not satisfied. The two-layer structure is tailored for the attributes as they are generally sparsely annotated for two reasons. First, it is common that the attributes of a product have not been fully provided by online retailers. Second, for some attributes (e.g., *color*), despite being annotated, only

one or a few states could be selected, but they can have many possible states. Therefore, if we treat the attributes equally to the categories (e.g., each attribute corresponds to a single task branch parallelized to the category branch), the whole network would be difficult to converge. However, concatenating them to a specific task branch will largely alleviate the difficulties. Our empirical study shows that the proposed architecture attains better results than the counterpart that treats the attributes task-by-task (see Section 4.3).

4 EXPERIMENTS

4.1 Dataset and Settings

We use a public dataset containing 971,467 real-world product images for evaluation. The dataset is released by the Alibaba large-scale image search challenge. All the images are picked out from *Taobao* such that they are described by the category hierarchy and attributes predefined by *Taobao;* for example, each image is associated with two categorical labels (i.e., root and leaf categories) and/or a few attribute labels (e.g., color, pattern, etc). The hierarchy consists of 10 root categories and 547 leaf categories. Since the categories are hierarchically partitioned and nonoverlapping at each layer, correct prediction on the leaf category also implies correct prediction on the root category. The dataset provides seven kinds of attributes. They are *shape*, *pants length*, *type of packages*, *sleeve length*, *pattern*, *brand*, and *color*. Each attribute has a varying number of possible states, and there are 160 states in total. As anticipated, the attributes are sparsely annotated. Although released for image retrieval competition originally, the dataset also constitutes a repository for evaluating large-scale product classification.

We performed a series of preprocessing steps on the dataset since images are not uniformly distributed among the leaf categories (as many as 99,351 and as few as 1). To ensure a reliable evaluation, we filter out leaf categories with less than 1,000 samples. As a result, we obtain 10 root and 129 leaf categories with 894,571 images in total, which compose the full dataset evaluated in this work. In addition, to better model the attributes, we remove images without any attribute labels from the full dataset, resulting a more compact core dataset containing 473,006 images. Similarly, images are not uniformly distributed among the attributes (as many as 224,167 and as few as 5,370). To avoid the use of too sparse attributes, we ignore the *type of packages* and *brand*, which are contained in less than 10,000 images. Possible states of the remaining five attributes are 104 in total.

Given the full and core datasets, we randomly split the images into training, testing, and validation sets category-by-category following the fixed partition ratios of 7:2:1. Note that the partition is based on the category and thus does not necessarily distribute the attributes at 7:2:1 for the three sets. As the evaluation metric, similar to the ImageNet classification competition [8], the top-1 and top-5 accuracies are adopted. Moreover, since the categories are hierarchically structured, we summarize the image-level accuracy according to two granularities. They are image-by-image accuracy (ACC) and mean accuracy (MACC), which calculates ACCs at the leaf category level.

4.2 Compared Approaches

Since we aim at leveraging structure-aware knowledge to improve classification performance, and existing studies in this direction are relatively few, we compare the following methods to quantitatively analyze the performance gains obtained from each of the efforts proposed in this work.

(1) bCNN: The basic CNN, which uses VGG-16 as the base network and sets the number of outputted nodes to 129; it merely uses the flat one-versus-all scheme to learn the model.

(2) ssCNN: The salient-sensitive CNN. It inserts a spatial attention module into the bCNN as described in Section 3.2, aiming to instruct the model to focus more on the product foreground.
(3) bCNN-MCR: It performs the MCR-based refinement on top of the paralleled bCNNs, where the two CNNs aiming to predict categorical labels at different granularities are carried out first. Then their prediction scores are adjusted by the MCR, whose weights are learned from a training set, as described in Section 3.3.
(4) ssCNN-MCR: It performs the MCR-based refinement on top of the paralleled ssCNNs. It is similar to bCNN-MCR except that the preceding CNNs are all ssCNNs rather than bCNNs.
(5) bCNN-MTL: Multitask deep learning built on top of bCNN, where VGG-16 has been modified to fit the multitask requirement (i.e., stacked with two or more private modules in parallel that learn different tasks). It has two different variants according to the tasks modeled (i.e., bMTL-RL that takes both root and leaf categories as the tasks, and bMTL-CA that takes both leaf categories and attributes as the tasks). In the experiment, we actually use the two variants for evaluation. Details of multitask learning are described in Section 3.4.
(6) ssCNN-MTL: Multitask deep learning built on top of ssCNN. It is similar to bCNN-MTL except that the preceding shared CNNs are all ssCNNs rather than bCNNs. Analogously, two variants, respectively termed ssMTL-RL and ssMTL-CA, are both provided for evaluation.

As for training details, similar to existing studies [36], we use Stochastic Gradient Descent (SGD) with backpropagation to train these models. Momentum and weight decay are set to 0.9 and 0.0005, respectively, for all models. As for the learning rate, it is initially set to 0.001 and decayed to 0.0001 after a fixed number of training iterations for bCNN and ssCNN, while initially set to 0.0001 and decayed to 0.00001 when preset iterations are reached for the MTL-based approaches. For bMTL-RL and ssMTL-RL, the parameters trading off the loss term (i.e., λ_i in Equation (5)) are empirically set to 0.1 for the root category and 1 for the leaf category, respectively, which are roughly inversely proportional to the number of root and leaf categories. For bMTL-CA and ssMTL-CA, they are set to 1 for the leaf category, 0.5 for *color*, 0.3 for *sleeve length*, and 0.1 for the remaining 3 attributes. The parameters are also roughly inversely proportional to the number of samples labeled with these attributes. Following the scheme in the ImageNet classification competition, all training images are augmented with random crop, mirror, and deformation to avoid overfitting. The augmented images are finally resized to 224*224 before being fed into the CNNs. All implementations are carried out on a workstation with one Titan XP GPU. It takes 26 to 32 hours to train one of the models on the full dataset using the Caffe platform.

4.3 Experimental Results and Analyses

In this section, we evaluate the aforementioned eight models on both full and core datasets, aiming to get insights about how the structure-aware knowledge should be utilized. Table 1 shows the accuracies of the eight methods on the two datasets. In the table, *top1-full* denotes the top-1 performance on the full dataset, and the rest are defined similarly. In the following, we will summarize the overall performance first, then analyze the results model by model to quantitatively validate our efforts. Since, for all these methods, differences in top-5 performance are often small (as they are already very high), it is meaningless to compare the methods on top of them. Thus, in the experiments, we mainly analyze the models based on their top-1 performance.

Overall performance. As can be seen, ssMTL-CA, the method that combines salient-sensitive CNN learning and multi-ask deep learning on top of both the leaf categories and attributes,

Table 1. Product Classification Performance (%) of the Eight Approaches on Both Full and Core Datasets

		bCNN	ssCNN	bCNN-MCR	ssCNN-MCR	bMTL-RL	bMTL-CA	ssMTL-RL	ssMTL-CA
ACC	top1-full	78.83	79.38	78.99	79.07	79.26	79.23	79.42	79.25
	top1-core	79.11	79.38	79.17	79.34	79.42	80.31	79.58	80.42
	top5-full	96.77	97.22	96.70	96.67	96.73	96.83	96.79	96.79
	top5-core	97.05	96.93	96.99	96.80	96.99	97.42	96.92	97.47
MACC	top1-full	71.09	71.58	71.16	71.59	72.46	72.02	72.60	72.12
	top1-core	60.94	63.50	61.56	63.51	63.39	68.62	63.63	68.34
	top5-full	94.44	94.44	94.25	94.27	94.61	94.66	94.64	94.65
	top5-core	90.49	91.65	89.54	89.44	91.28	92.83	91.71	93.78

achieves the best performance among the eight models in terms of both ACC and MACC in most cases. Compared with the bCNN counterpart, the improvements are 0.5% and 1.7% in terms of the top-1 ACC, and 1.4% and 12.1% in terms of the top-1 MACC on full and core datasets, respectively. The relatively small improvements on the full dataset occur because the attributes are quite sparse in that dataset, which limits the validity of attribute modeling. We also see that the results obtained from different evaluation metrics are quite different. Compared with MACC, ACC generally has higher absolute results but much smaller improvement gains among the compared models. This is because, first, it is partially caused by the uneven distribution of images across the categories. More data often mean better model training, thus relatively higher results are obtained for large categories (i.e., with many samples) and relatively lower results for small categories. As a consequence, ACC is higher than MACC in absolute values. Second, we have proposed efforts to build more advanced CNNs and get more powerful feature representation. Therefore, the phenomenon of sample scarcity is alleviated somewhat and relatively larger performance gains are obtained for small categories, which results in more prominent improvements in terms of MACC. It is also observed that bMTL-CA attains quite similar results with ssMTL-CA, both of which are consistently better than bMTL-RL and ssMTL-RL, respectively. The comparisons not only show that the contribution mainly comes from the multitask architecture, but also illustrate the effectiveness of combining heterogeneous categories and attributes.

Incorporation of the salient-sensitive learning. As listed, there are four pairs of values that reflect the differences of with and without the salient-sensitive learning with respect to each evaluation metric and dataset. The salient-sensitive versions perform better than their bCNN counterparts in 15 out of the 16 cases under different settings when taking the top-1 performance as the evaluation metric. The results clearly reveal the necessity of incorporating salient-sensitive learning. It indeed assists the CNNs to pay more attention to the product foreground. However, it is also observed that the improvements are somewhat limited in some cases (e.g., multitask learning scenarios), partially attributed to the complexity of real-world product images. Meanwhile, powerful CNN models are also capable of implicitly doing attention learning, and, on the other hand, they weaken the effectiveness of the proposed salient-sensitive module to some extent.

Incorporation of the MCR. In Ai et al. [1], it is observed that the MCR-based refinement could lead to a few performance gains by using AlexNet as the base network. However, the conclusion is not consistently observed in Table 1. Specifically, the improvements are marginal, or even slight deteriorations are observed under different conditions. This might be explained because the role of MCR is to perform data fitting on a training set, and it is encoded as a weight matrix to guide the adjustment of prediction scores at the testing stage. MCR is a shallow learning method that is easily disturbed by noisy real-world data. It thus risks incorrectly encoding correlations among the

Table 2. Product Classification Performance (%) of Different Attributes on Different Task Branches

attribute	sleeve length	pants length	shape	color	pattern	all
ACC-branA	87.72	85.74	57.27	62.48	68.19	-
ACC-branC	80.39	80.34	80.13	80.31	80.20	80.42
MACC-branC	67.80	68.79	68.28	68.42	68.73	68.34

categories. In case of using AlexNet as the base network, the CNN is not as powerful as VGG-16. MCR could consistently improves the performance, while in case of using VGG-16, more complicated patterns are directly discovered by the network itself. The encoding of MCR is even noisier. The consequence of incorporating MCR is more unpredictable.

Incorporation of the multitask learning. Multitask deep learning provides more flexibilities on encoding correlations. Table 1 reports the results of the four multitask learning methods by assembling the building blocks above, where performance gains are obtained in a majority of cases when compared with their bCNN, ssCNN, and MCR-based counterparts. The results indicate that multitask learning methods can implement more effective product classification by employing a single end-to-end trainable CNN to encode different structure-aware clues as a whole. Furthermore, it is observed that the performance gains obtained from simultaneously modeling root and leaf categories are not as obvious as those from simultaneously modeling leaf categories and attributes. This is no surprise, as predicting categories at different granularities is somewhat homogeneous. Classifying root categories (with much fewer categories) is much easier than classifying leaf categories. Therefore, little knowledge could be gained from the root branch when learning the leaf branch. On the contrary, attribute prediction is a task quite different from category classification. It carries more complementary information and thus leads to more prominent improvements.

Incorporation of the attributes. Since image classification with the existence of multiple attributes constructs a less studied research scenario, we also analyze the performance of using the attributes jointly and individually in the multitask deep learning framework. To this end, we construct five multitask CNNs; each takes predicting one of the five attributes as the task in the attribute branch, using the ssMTL-CA architecture. In each CNN, besides the categorical loss, only the loss from the corresponding attribute is taken into account. Classification results on different branches are summarized in Table 2, where ACC-branA denotes the top-1 ACC in the attribute branch and ACC-branC denotes the top-1 ACC in the leaf category branch. As can be seen, the performance is quite different when inspecting from the attribute branch. The *sleeve length* and *pants length* attributes give satisfactory performance. We can explain this from two aspects. First, the number of images labeled by the two attributes is not too few, which brings relatively adequate samples for model learning. Second, the two attributes are strongly category-sensitive (i.e., they carry distinct information that tells us what categories they should be classified to). Moreover, we also note that there are many samples labeled with the *color* attribute. However, its performance is not as good as the previous two attributes. This phenomenon again shows the challenge of attribute classification on real product images when attributes are sparsely provided.

On the other hand, despite varying significantly in the attribute branch, the performance of the five individually learned CNNs and the jointly learned CNN on leaf category classification show much smaller variance in both ACC and MACC. This implies that, for the proposed mult-task CNN, what's important is introducing a dedicated attribute prediction branch, which carries quite different information for the shared layers and leads to better category classification. Compared

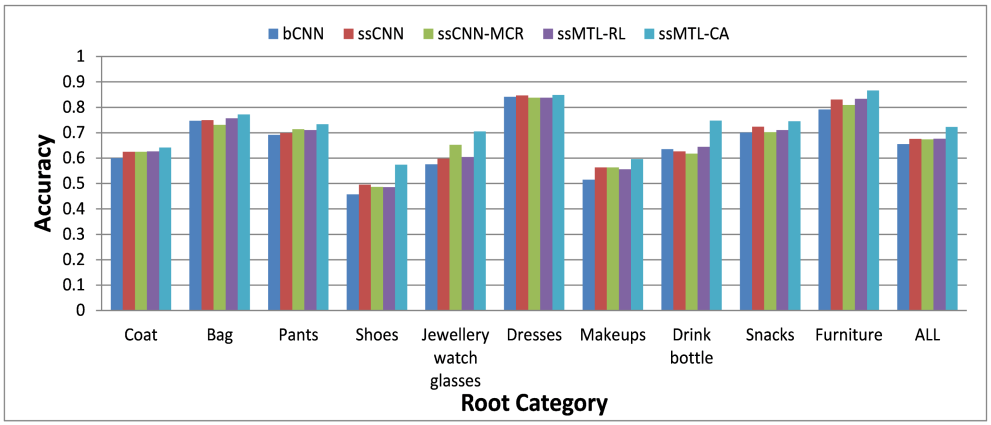

Fig. 5. Classification performance of different methods on core dataset aggregated to the 10 root categories, where each bar is the averaged MACC of a root category with respect to a method.

with training based on a single attribute, using multiple kinds of attributes to train the model just means involving more training samples. The observation is not in accordance with our anticipations. Perhaps it is mainly due to the severe absence of attribute labels. Not containing an attribute label does not necessarily mean that an image does not have this attribute. Therefore, learning within such a noisy circumstance somewhat limits the ability of deeply using the knowledge conveyed by attributes.

Since we are also interested in inspecting the performance on a macro scale, we group the classification results on the core dataset into the 10 root categories. Figure 5 shows the results, where each bar is the averaged MACC of a root category with respect to a method. It is different from the values reported in Table 1, where the micro averages over images or leaf categories are reported, here we give the macro average over the root categories. The main observations are:

- ssMTL-CA still attains the best overall performance over the 10 root categories. A prominent performance gain of 10.3% is observed when comparing ssMTL-CA with its bCNN counterpart, indicating again that salient-sensitive modeling and multitask deep learning on top of both categories and attributes are reasonable ways to leverage structure-aware knowledge.
- The categories present different classification difficulties. It is observed that *furniture*, *dresses,* and *bag* generally attain better classification performance than other categories. We argue that this is because their visual patterns are not as diverse as other categories. On the contrary, the performance of *makeup* and *jewellery watch glasses* is somewhat low. This is mainly because products in those categories are not only more diverse in visual appearance, but also small in size, with larger variants in quantity and the capturing view, all of which are challenges for visual-based classification protocols.
- ssMTL-CA shows large improvement gains in categories *shoes*, *jewellery watch glasses,* and *drink bottle*. We argue that this is largely attributable to salient-sensitive learning. A large portion of these products are small in size and thus could benefit more from a dedicated salient module. On the other hand, for popular categories like *coat*, *bag*, and *pants*, the gains are not as prominent as previously, mainly because they are large categories and thus their models are better trained. This neutralizes the differences from different methods to some extent.

Fig. 6. Classification examples of ssMTL-CA, where errors are marked by a red rectangle.

In Figure 6, we also depict some classification examples from ssMTL-CA including both correctly and wrongly classified images; the latter ones are marked in red by a rectangle. It is seen that the proposed ssMTL-CA could still make reliable classifications in challenging cases, such as cluttered backgrounds and lots of background text, as well as on artificial images such as picture-in-picture and spliced images. For the wrongly classified examples, we analyze them carefully and group them into two main causes. One is that the image is too challenging. For example, in the leftmost image in the second row in Figure 6, the sample is rather difficult such that even a human cannot distinguish it well because of the overlaid text. However, for product images, adding text and/or logos is a commonly applied editing technique. Therefore text/logos are likely to be interpreted as noise by the classifiers as they are usually category insensitive. The other cause occurs when multiple objects belonging to different categories are simultaneously displayed in an image; for example, it is clearly seen that a *bag* and a *glass* are shown in the fourth image of the first row in Figure 6. The image is labeled as *glass*, but we classify it as *bag*. Strictly speaking, this image should not be regarded as a wrongly classified case as the *bag* indeed appears. Some images naturally have products coming from multiple categories. Perhaps it is more suitable to assign multiple categorical labels at the same layer, but this is likely to challenge the current policy of online retailers (i.e., a product image that only belongs to one category in a layer). These two kinds of problems are somewhat unique, and how to tackle them presents interesting issues worthy of further investigation.

Drawing from the above observations, we come to the conclusion that correlations among both categories and attributes are indeed useful in product classification. It is better to use the multi-task deep learning framework to combine these structure-aware clues. Moreover, by additionally adding a dedicated salient-sensitive module, the classification model will be even powerful in most cases. A total of a 12.1% performance gain on MACC is obtained by leveraging these structure-aware clues.

5 CONCLUSION

Predicting the categorical labels of product images is an essential classification task. It supports the services of online retailers from a number of aspects and is thus of crucial importance. Beyond the use of powerful CNN models to implement this task, in this work, we emphasize leveraging structure-aware knowledge, including correlations among categories and descriptive attributes, to further enhance classification accuracy. To this end, three efforts covering the salient-sensitive CNN, MCR-based refinement, and multitask deep learning are proposed. Experimental results on

nearly 1 million *Taobao* images basically validate our proposals, from which performance improvements are observed when quantitatively analyzing the three efforts individually and jointly. The ssMTL-CA model built on top of both categories and attributes achieves the best performance, where as high as 12.1% performance gains on MACC are attained compared with its bCNN counterpart. The results are encouraging in that it verifies that correlations among categories and attributes are indeed helpful in large-scale classification, and our explorations give valuable insights into how to leverage this knowledge.

Despite the improvements obtained, studies on the use of multiple structure-aware knowledge affiliated with images to assist classification are still in their infancy. There is still plenty of work ahead. First, the proposed approaches confront the fact that performance gains are still limited; we thus are interested in exploring more effective ways of utilizing these categories and attributes. Second, we merely discuss the prediction of categories in this work. However, predicting attributes is also a meaningful task as it is observed that attributes are commonly sparsely labeled. Richer annotations can lead to better product management, and thus we plan to work on this problem. Third, besides the *Taobao* images, we also would like to collect more product images from other online retailers such as *Amazon*, *Jingdong*, and more and use them as testbeds to further validate our methods.

REFERENCES

[1] Shanshan Ai, Caiyan Jia, and Zhineng Chen. 2017. Large-scale product classification via spatial attention based CNN learning and multi-class regression. In *Proceedings of the International Conference on Multimedia Modeling, Reykjavik, Iceland*. Springer, 176–188.

[2] Jinfeng Bai, Zhineng Chen, Bailan Feng, and Bo Xu. 2014. Image character recognition using deep convolutional neural network learned from different languages. In *Proceedings of the International Conference on Image Processing, Paris, France*. IEEE, 2560–2564.

[3] Lunshao Chai, Zhen Qin, Honggang Zhang, Jun Guo, and Christian R Shelton. 2012. Re-ranking using compression-based distance measure for content-based commercial product image retrieval. In *Proceedings of the International Conference on Image Processing, Lake Buena Vista, Orlando, FL*. IEEE, 1941–1944.

[4] Jingjing Chen and Chong-Wah Ngo. 2016. Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the 2016 ACM Conference on Multimedia, Amsterdam, Netherlands*. ACM, 32–41.

[5] Zhineng Chen, Juan Cao, Tian Xia, Yicheng Song, Yongdong Zhang, and Jintao Li. 2011. Web video retagging. *Multimedia Tools and Applications* 55, 1 (2011), 53–82.

[6] Zhineng Chen, Chong-Wah Ngo, Wei Zhang, Juan Cao, and Yugang Jiang. 2014. Name-face association in web videos: A large-scale dataset, baselines, and open issues. *Journal of Computer Science and Technology* 29, 5 (2014), 785–798.

[7] Zhineng Chen, Wei Zhang, Bin Deng, Hongtao Xie, and Xiaoyan Gu. 2019. Name-face association with web facial image supervision. *Multimedia Systems* (2017).

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, Florida*. IEEE, 248–255.

[9] Jianlong Fu, Heliang Zheng, and Tao Mei. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA*. 4476–4484.

[10] Marian George and Christian Floerkemeier. 2014. Recognizing products: A per-exemplar multi-label image classification approach. In *Proceedings of the European Conference on Computer Vision, Zurich, Switzerland*. Springer, 440–455.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH*. 580–587.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada*. 770–778.

[13] Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA*. 4700–4708.

[14] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. 2016. Part-stacked CNN for fine-grained visual categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada*. 1173–1182.

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. 2015. Spatial transformer networks. *Neural Information Processing Systems* (2015), 2017–2025.
[16] Yugang Jiang, Jun Yang, Chongwah Ngo, and Alexander G. Hauptmann. 2010. Representations of keypoint-based semantic concept detection: A comprehensive study. *IEEE Transactions on Multimedia* 12, 1 (2010), 42–53.
[17] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. 2018. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 2 (2018), 352–364.
[18] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. 2011. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proceedings of the CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, Vol. 2. 1.
[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Neural Information Processing Systems, Lake Tahoe, Nevada.* 1097–1105.
[20] Hao Lei, Kuizhi Mei, Jingmin Xin, Peixiang Dong, and Jianping Fan. 2016. Hierarchical learning of large-margin metrics for large-scale image classification. *Neurocomputing* 208 (2016), 46–58.
[21] Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees G. M. Snoek, and Alberto Del Bimbo. 2016. Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Computing Surveys* 49, 1 (2016), 14:1–14:39.
[22] Zhetao Li, Jie Zhang, Kaihua Zhang, and Zhiyong Li. 2018. Visual tracking with weighted adaptive local sparse appearance model via spatio-temporal context learning. *IEEE Transactions on Image Processing* 27, 9 (2018), 4478–4489.
[23] Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. 2015. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts.* IEEE, 1666–1674.
[24] Tsungyu Lin, Aruni Roychowdhury, and Subhransu Maji. 2015. Bilinear CNN models for fine-grained visual recognition. In *Proceedings of the International Conference on Computer Vision, Santiago, Chile.* 1449–1457.
[25] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada.* 1096–1104.
[26] Shiyang Lu, Tao Mei, Jingdong Wang, Jian Zhang, Zhiyong Wang, and Shipeng Li. 2015. Exploratory product image search with circle-to-search interaction. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 7 (2015), 1190–1202.
[27] Changzhi Luo, Zhetao Li, Kaizhu Huang, Jiashi Feng, and Meng Wang. 2018. Zero-shot learning via attribute regression and class prototype rectification. *IEEE Transactions on Image Processing* 27, 2 (2018), 637–648.
[28] Tiendung Mai, Thanh Duc Ngo, Duydinh Le, Duc Anh Duong, Kiem Hoang, and Shinichi Satoh. 2017. Efficient large-scale multi-class image classification by learning balanced trees. *Computer Vision and Image Understanding* 156 (2017), 151–161.
[29] Yingwei Pan, Ting Yao, Houqiang Li, Chong-Wah Ngo, and Tao Mei. 2015. Semi-supervised hashing with semantic confidence for large scale visual search. In *Proceedings of the ACM SIGIR Conference, Santiago, Chile.* ACM, 53–62.
[30] Florent Perronnin and Diane Larlus. 2015. Fisher vectors meet neural networks: A hybrid classification architecture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts.* IEEE, 3743–3752.
[31] Zhaofan Qiu, Yingwei Pan, Ting Yao, and Tao Mei. 2017. Deep semantic hashing with generative adversarial networks. In *Proceedings of the ACM SIGIR Conference, Tokyo, Japan.* ACM, 225–234.
[32] Scott E Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada.* 49–58.
[33] Jorge Sánchez and Florent Perronnin. 2011. High-dimensional signature compression for large-scale image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO.* IEEE, 1665–1672.
[34] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv: Computer Vision and Pattern Recognition* (2013).
[35] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations* (2015).
[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts.* IEEE, 1–9.

[37] Dequan Wang, Zhiqiang Shen, Jie Shao, Wei Zhang, Xiangyang Xue, and Zheng Zhang. 2015. Multiple granularity descriptors for fine-grained categorization. In *Proceedings of the International Conference on Computer Vision, Boston, Massachusetts*. IEEE, 2399–2406.

[38] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-UCSD birds 200. *California Institute of Technology* (2010).

[39] Qiong Wu and Pierre Boulanger. 2016. Enhanced reweighted MRFs for efficient fashion image parsing. *ACM Transactions on Multimedia Computing, Communications, and Applications* 12, 3 (2016), 42.

[40] Hongtao Xie, Ke Gao, Yongdong Zhang, and Jintao Li. 2011. Local geometric consistency constraint for image retrieval. In *Proceedings of the International Conference on Image Processing, Belgium, Brussels*. IEEE, 101–104.

[41] Hongtao Xie, Yongdong Zhang, Jianlong Tan, Guo Li, and Jintao Li. 2014. Contextual query expansion for image retrieval. *IEEE Transactions on Multimedia* 16, 4 (2014), 1104–1114.

[42] Lexing Xie, Rong Yan, Jelena Tešić, Apostol Natsev, and John R. Smith. 2010. Probabilistic visual concept trees. In *Proceedings of the 18th ACM international Conference on Multimedia, Firenze, Italy*. ACM, 867–870.

[43] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *Computer Science* (2015), 2048–2057.

[44] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis Decoste, Wei Di, and Yizhou Yu. 2015. HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts*. IEEE, 2740–2748.

[45] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada*. 21–29.

[46] Ting Yao, Fuchen Long, Tao Mei, and Yong Rui. 2016. Deep semantic-preserving and ranking-based hashing for image retrieval. In *Proceedings of the International Joint Conferences on Artificial Intelligence, New York, NY*. 3931–3937.

[47] Chunjie Zhang, Jian Cheng, and Qi Tian. 2018. Image-level classification by hierarchical structure learning with visual and semantic similarities. *Information Sciences* 422 (2018), 271–281.

[48] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. 2016. Picking deep filter responses for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada*. 1134–1142.

[49] Shiai Zhu, Xiaoyong Wei, and Chong-Wah Ngo. 2014. Collaborative error reduction for hierarchical classification. *Computer Vision and Image Understanding* 124 (2014), 79–90.

Received November 2017; revised April 2018; accepted June 2018